Proportional Increase Multiplicative Decrease (PIMD) Wireless Scheduler: An Efficient Scheduler for IEEE 802.11e HCF

Arshad Hussain and Shaban Qaiser

Department of Telecommunication and Computer Engineering,
National University of Computer & Emerging Sciences
Lahore, Pakistan
`{arshad.hussain,shaban.qaiser}@nu.edu.pk`

Abstract. In this paper, we propose a new wireless scheduling algorithm for the IEEE 802.11e HCF. The algorithm grants the mobile stations variable time for the upstream data flow in proportion to the queue size of the transmission buffer. At the same time, it retrieves half of the extra time allocated in previous cycle from those flows whose requirement has stopped. Hence, the system achieves stability by preventing the ping-pong phenomena. The algorithm is computationally simple, and as compared to the other algorithms, it gives bounded delays and jitter to the real-time applications under heavy load conditions.

1 Introduction

Wireless local area networks (WLANs) have provided an edge over wired networks in ease of installation, increased bandwidth and decreased prices. They hold the promise of providing unprecedented mobility, flexibility and scalability. The need for mobile computing has launched a successful market for WLAN promising to replace most of the wired LAN infrastructures in the near future, allowing users to roam inside a building or a university without interrupting their communication sessions and avoiding the use of cables.

The advancement in digital technology has also led to an increased demand in multimedia applications such as video conferencing, video on demand, voice chat, and other time-bounded data transfer activities over the networks. However, wireless networks also exhibit higher bit-error rates and fluctuating dynamic bandwidth. Hence, these applications necessitate that both wired and wireless networks provide such services. It makes the quality of service (QoS) provisions a critical issue. QoS is the capability of a network to provide resource assurance and service differentiation during the connection. Currently, IEEE 802.11 WLAN standard is being deployed widely and rapidly for its simplicity and robustness against failures. However, the widespread use of multimedia networking applications has created the need for WLANs to support real time services with stringent QoS requirements. The development of IEEE 802.11e can be considered as an effort in this direction. It is a new standard which defines QoS at MAC level of WLAN.

C. Wang (Ed.): AccessNets, LNICST 6, pp. 323–332, 2009.

Currently, various models are being developed in order to provide QoS to real time applications in WLANs. The core architecture of QoS models comprises efficient real-time scheduler and call admission control (CAC) mechanism. They are deployed in the centralized station called QoS access point (QAP) when WLAN is operating in infrastructure mode. The scheduler provides transmission opportunities to mobile stations so that their time-critical QoS requirements can be fulfilled. The QoS received by different applications running on mobile nodes depends on the efficient working of the scheduler, while CAC ensures that the scheduler is not overloaded, statically or statistically. Our research effort is aimed at developing a dynamic scheduling mechanism for IEEE 802.11e MAC. The motivation for this work comes from the fact that IEEE 802.11e specifies a sample scheduler in its standard, yet it leaves the door open for the development of more efficient scheduling mechanisms. The paper has been divided in the following sections. In section 2, we give a brief introduction of QoS, IEEE 802.11e, and the work done so far in developing wireless schedulers for QoS-WLAN. In section 3, we propose PIMD wireless scheduler and discuss its characteristics followed by its simulation comparison with most recognized FHCF scheduler [14] in section 4. In the last section, we conclude our work and give direction for further enhancement.

2 Background and Literature Survey of Wireless Scheduler

2.1 QoS and Wireless Scheduling

Broadband networks provide QoS differentiation and guaranteed services for heterogeneous classes of traffic with different QoS requirements, e.g., throughput, bit error rates, delay, jitter, packet loss rate etc. QoS support is dictated by the service models such as IntServ [1] for per-flow-based guaranteed QoS performance by pre-allocation of resources or DiffServ [2] for class-based QoS support. By using the QoS service primitives, the application provides the QoS-network with traffic specifications (TSPEC) that makes the network adapt to the traffic in an optimized manner. Since the most precious resource in wireless networks is the bandwidth, an efficient and optimized utilization of the bandwidth results in greater channel utilization and higher throughput. As the mobile host moves, channel parameters like the available bandwidth and bit error rate change dynamically due to the intrinsic nature of the wireless medium. An efficient wireless scheduling algorithm aims at minimizing the unproductive transmissions on erroneous links, and at the same time, maximize the effective service delivered and the utilization of the wireless channels. The challenges faced in designing schedulers for wireless networks have been described in [3], [4]. A good centralized scheduling algorithm should be designed so that minimal numbers of scheduling-related control messages are required to be sent from the mobile hosts in order to conserve limited battery power. The scheduling algorithm should be computationally efficient so that it can be executed at high speed to schedule real-time multimedia traffic with stringent timing requirements. Development of scheduling mechanisms for real-time applications is an active area of research. Its basic architecture has been modeled mathematically and characteristics of good schedulers have been defined in [6]. Also, many wireless schedulers have been compared with each other in terms of channel utilization, delay bound, fairness, and complexity in survey reports [3], [4], [6]

2.2 Wireless Schedulers for IEEE 802.11e WLAN

IEEE 802.11e standard [5] defines a MAC enhancement to provide QoS in WLANs. The basic service set (QBSS) consists of a QAP and mobile stations (QSTAs). QAP is usually a centrally located, fixed station with additional responsibilities of managing the WLAN in infrastructure mode. All QSTAs in the QBSS associate themselves with the QAP in order to communicate with the outside world. IEEE 802.11e introduces hybrid coordination function (HCF), which determines when a QSTA is permitted to transmit and receive MAC service data units (MSDU). HCF works by using two channel access mechanisms; enhanced distributed channel access (EDCA) and HCF controlled channel access (HCCA). The EDCA is a contention based channel access method which is used for prioritized QoS (DiffServ) during contention period (CP). The HCCA is used for parameterized QoS (IntServ) during both contention period (CP) and contention free period (CFP). HCCA is a polling based access mechanism in which the medium access is governed by the QAP when the network is set up in infrastructure mode. The HCF gets control of the channel after every SI. It can transmit downstream data by allocating TXOP to QAP or poll the QSTAs by giving TXOPs to their admitted traffic streams (TSs). A polled QSTA can transmit multiple MAC frames in a single TXOP provided it does not violate the TXOP limit. A traffic stream (TS) requiring certain parameterized QoS guarantees is described by a set of traffic specifications (TSPEC) which includes nominal MSDU size (L), maximum MSDU size (M), minimum service interval (SI_{min}), maximum service interval (SI_{max}), mean application data rates (ρ), maximum burst size, and peak data rate. The QSTA requests QAP for initialization of individual TSs (uplink, downlink or bi-directional). The call admission control (CAC) unit at QAP decides whether to include it in the polling list or not. Once TS has been admitted, it is the responsibility of QAP to poll the TS, based on its TSPEC. The QAP uses a centralized deadline scheduler to decide which TS to poll and when to poll it. It computes the duration of TXOP that is to be granted to the TSs and polices and shapes the amount of traffic to provide fairness among TSs. HCCA allows better channel utilization while maintaining low jitter and latency for high priority QoS streams.

The IEEE 802.11e standard has given a sample scheduler [5, Annex K]. It uses mandatory TSPEC and allocates constant TXOP for each traffic stream. The scheduler works for constant bit rate (CBR) applications but it doesn't incorporate variations in actual transmission rate, application data rate and packet size. Therefore, the transmission of variable bit rate (VBR) and traffic with occasional long bursts can lead to significant transmission delays, increased average queue length and packet drops. Grilo's scheduler [7] allocates variable TXOPs and polls STAs in variable time and service intervals. The variable TXOP is made available by using token bucket, and earliest due date (EDD) is used to select the flow from the subset of eligible flows. The Grilo Scheduler is more flexible and attains better performance with respect to delay constraints at the cost of increased complexity. Allocating variable TXOPs makes it more suitable for VBR applications. The Extended Grilo Scheduler incorporates the queue size and TXOP duration requested field, and traffic requirements in Grilo algorithm for the computation of next TXOP.

The fair HCF (FHCF) scheduler [8] uses the queue size of upstream TS sent by a non-QAP QSTA at the end of its TXOP finish time to predict its TXOP for the next *SI*. It also predicts the ideal queue size for the TS at the start of the next *SI* using TSPEC. This difference is used to compute the TXOP for the next *SI*. The computation also takes into account the moving average of the absolute difference of the predictions in the previous *SI*'s in form of adding a correction term in the next TXOP. The FHCF scheme supports variable application data rates and /or packet sizes. The stochastic nature of VBR and bursty traffic is taken care of by averaging the previous errors in queue size estimation. FHCF admits more flows in HCCA as compared to others, and it achieves a higher degree of fairness among various multimedia flows while supporting bandwidth and delay requirements for a wide range of network loads. However, it is biased towards increasing TXOPs for TSs and it is computationally very expensive.

There are many other schedulers which have been proposed for IEEE 802.11e. Grag et al. [9], [10] have proposed a simple scheduler for IEEE 802.11e based on the queue lengths and priority of the traffic streams at the STAs. Perfectly Periodic Scheduling [11] for 802.11e home networks as an efficient way to reduce the access waiting time based on probability usage. Feedback based dynamic scheduler (FBDS) [12] is a queue based algorithm in which the HC assigns TXOPs to each TSs such that each queue is drained during the next controlled access period (CAP). It employs a discrete time linear feedback model and proper disturbance model for predicting queue size at the non-AP QSTAs. It is an efficient scheduler as it achieves the desired delay asymptotically but its complexity is very high and it admits fewer flows. The stability of the feedback system is also a crucial issue.

3 Proportional Increase Multiplicative Decrease (PIMD) Wireless Scheduler for IEEE 802.11e

In this section, we propose a simple scheduler at QAP which dynamically allocates the TXOP based on actual data which is being generated by the application at the mobile nodes. In this scheme, the QAP polls all STAs with admitted TSs in one CAP, and uses the mandatory TSPEC to compute the initial TXOP for individual TSs. If the polled flow has transmitted the queue size of its non-transmitted data in the buffer at the end of polled TXOP, PIMD scheduler increases the TXOP of a flow dynamically in proportion to queue size in the next service interval. Once the data has been drained, the scheduler reduces the extra TXOP exponentially in subsequent *SI*'s in order to keep the system stable.

3.1 Initialization of PIMD

When a stream requests initialization, it also transmits QoS requirement parameter set (ρ, L, SI_{max}, M, R) to the QAP in TSPEC. QAP uses the sample scheduler to allocate initial TXOPs to all flows by computing the schedule service interval (*SI*) for one CAP, and the TXOP (T_i) for all stations within *SI* using the following formulae;

$$SI \leq \min_{i} \left\{ SI_{\max,i} \right\} \tag{1}$$

$$T_i(1) = \max\left[\frac{\rho_i \times SI}{R_i}, \frac{M}{R_i} \right] + O \tag{2}$$

where O is physical layer overhead.

3.2 The PIMD Algorithm

During any service interval repetition (n=1, 2, 3…), if QAP receives a non-zero queue size $q_i(n)$ from a flow in its last upstream polled frame, the scheduler allocates additional TXOP to the flow in the next *SI*. We denote the extra TXOP by $\delta_i(n)$. The unallocated portion of the *SI* is divided among all such flows proportionately at the end of *SI* (proportional increase). However, if due to the additional allocation, the flow's buffer gets empty, $\delta_i(n)$ is not reduced to zero in the next SI; it is decreased by half (multiplicative decrease).

In order to model the algorithm, we define Ω as the set of all admitted flows. Let $D_{PI}(n)$, $D_{MD}(n)$, and $D_{NC}(n)$ be the three disjoint subset of Ω defined for *SI*(n) as below;

$$\begin{aligned} &D_{PI}(n) = \{\forall i : q_i(n) > 0\}. \\ &D_{MD}(n) = \{\forall i : \delta_i(n-1) \neq 0, q_i(n) = 0\}. \\ &D_{NC}(n) = \{\forall i : \delta_i(n-1) = 0, q_i(n) = 0\}. \end{aligned} \tag{3}$$

It is easy to see that

$$\Omega = D_{PI}(n) \cup D_{MD}(n) \cup D_{NC}(n). \tag{4}$$

Hence, $D_{PI}(n)$, $D_{MD}(n)$ and $D_{NC}(n)$ form partitions of Ω. It may be noted that number of flows in these partitions may vary at every repetition of SI(n). The TXOP $T_i(n+1)$ is allocated to every admitted flow in the following steps.

Step I (Multiplicative Decrease)

$$\delta_i(n+1) = \begin{cases} \delta_i(n)/2, & \forall i \in D_{MD}(n) \\ 0, & \forall i \in D_{NC}(n) \end{cases} \tag{5}$$

$$T_i(n+1) = T_i(n) + \delta_i(n+1).$$

Step II (Proportional Increase)

$$SI_{free}(n+1) = SI - \sum_{j \in D_{PI}(n)} T_j(n) - \sum_{j \in \Omega - D_{PI}(n)} T_j(n+1) \tag{6}$$

$$\delta_i(n+1) = SI_{free}(n+1) * q_i(n) / \sum_{j \in D_{PI}(n)} q_j(n), \qquad \forall i \in D_{PI}(n)$$

$$T_i(n+1) = T_i(n) + \delta(n+1).$$

3.3 Discussion

The PIMD scheduler provides long-term fairness and its computational complexity is extremely low. However, in implementing PIMD algorithm, the multiplicative decrease (MD) always precedes proportional increase (PI). This way, the extra bandwidth accumulated due to non-requirement of certain TSs during MD step is fairly allocated among the TSs with non-zero queue sizes.

4 Experimental Result

The PIMD algorithm has been tested by simulating it on the Network Simulator-2 (NS2) platform [13]. NS2 is an open-source public domain simulation tool that runs on Linux, and it is used for simulating local and wide area networks. FHCF 802.11e implementation by Pierre Ansel, Qiang Ni and Thierry Turletti has been used as the base of PIMD algorithm implementation in NS2 [14].

The algorithms have been tested for three types of traffic flows, namely, high priority audio (ON-OFF keying), medium priority VBR (H.261), and relatively low priority CBR (MPEG 4). The audio flows are implemented by an exponentially distributed traffic source generator with 400ms on time and 600ms off time. The traffic files provided by [8] have been used for VBR where the data rate varies between 170 Kbps to 240 Kbps in extreme cases. CBR traffic was generated with inter-arrival time of packets of 2ms. These traffic flows are mapped onto the traffic streams (TS) of a station. The basic traffic characteristics and the MAC parameters used in running TSs over IEEE 802.11e have been summarized in the Table 1.

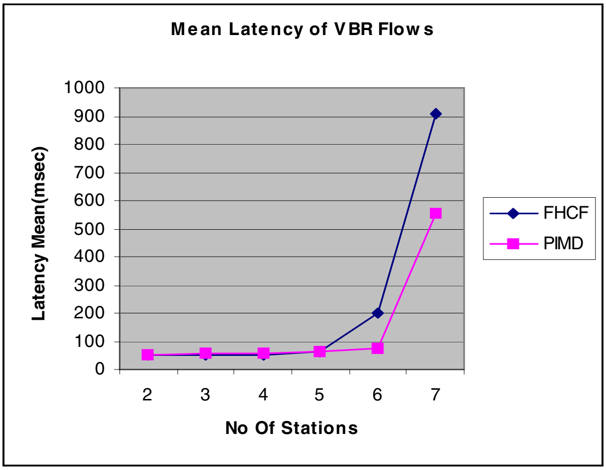
The PIMD algorithm has been tested for two scenarios. In the first scenario, it was tested only for VBR traffic on 2 Mbps channel with HCCA capturing the 90% of the duration. In the second scenario, we have tested it by scheduling three different types of traffic, with only one type of traffic flow running on one station. The channel bandwidth in 2nd scenario was set to 36Mbps and HCCA duration was 98%, so purely HCCA is tested. In both scenarios, the numbers of stations were increased from 2 to 7 for each type of flow which equivalently amounts to 26—92% increase in load. The HCCA load calculation also included the overhead for each packet to be transmitted. The performance of PIMD has been compared with FHCF in terms of mean delay and jitter as the load was increased.

Table 1. Traffic Flows and MAC Parameters

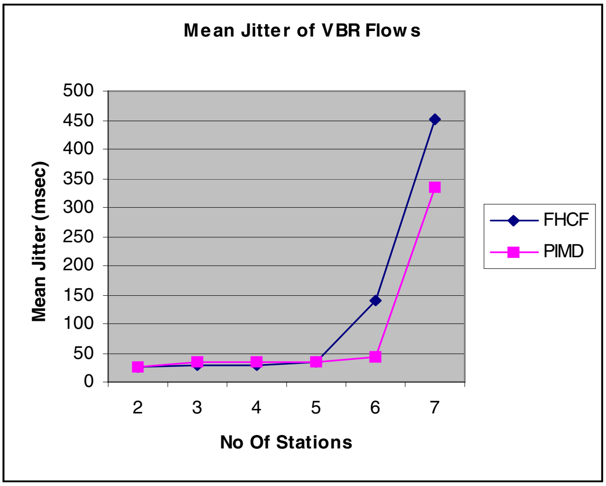
Traffic Type	**Mean data rate (kbps)**	**Mean MSDU size (bytes)**	**EDCA**			**HCCA**
			CW_{min}	CW_{max}	*Priority*	SI_{max} *(ms)*
Audio	64	160	7	15	6	50
VBR	200	660	15	31	5	100
CBR	3200	800	15	31	4	100
ALL	EDCA Long Retry Limit= 4, EDCA Short Retry Limit=7					

4.1 Test Scenario 1: Only VBR Traffic

In the first scenario, both PIMD and FHCF were tested for variable bit rate (VBR) traffic only, and the results have been plotted in Fig. 1. Under light load, performance of both schedulers was similar. However, under heavy load, PIMD outperformed FHCF by a wide margin. For example, at the load of 6 stations, the delay in FHCF was beyond tolerable range (201 ms) where as in PIMD, it was within acceptable bound (74 ms), i.e., one-third of FHCF (Fig. 1a). The delay measurement for both schedulers at the load of stations load was unacceptable; yet, the delay in the case of



(a) Delay



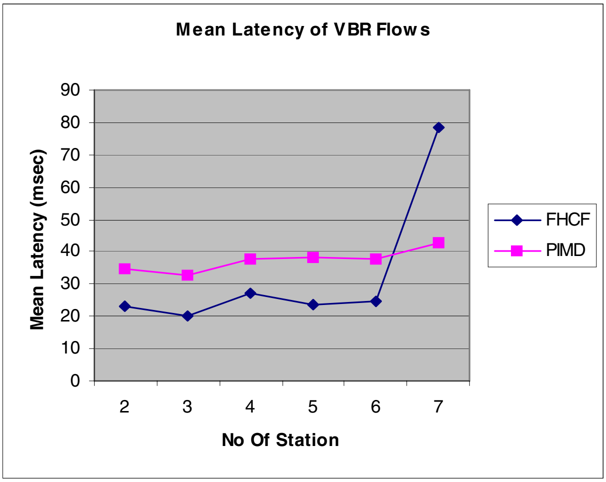
(b) Jitter

Fig. 1. PIMD vs. FHCF for Scenario I (a) Delay, (b) Jitter

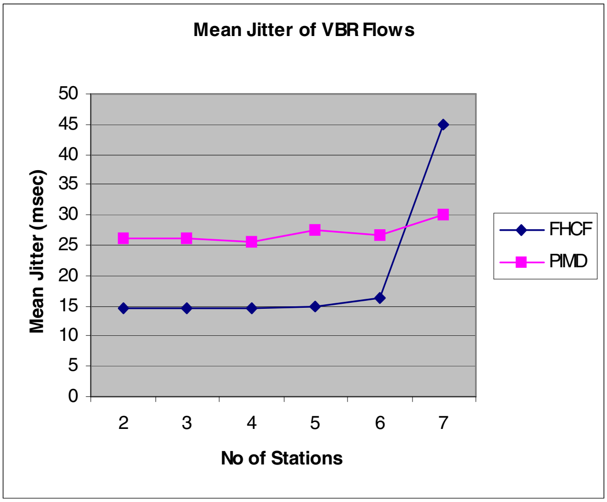
PIMD was half as compared to that due to FHCF. A similar trend has been observed in jitter (Fig. 1b).

4.2 Test Scenario 2: Simultaneous Audio, VBR, and CBR Traffic

In this scenario the scheduling algorithms were tested for isolated traffic. Three types of isolated traffic were simulated (audio, variable bit rate and constant bit rate) with each of the traffic flow running on a separate node independently. Few nodes had only audio traffic running over them, while few had VBR traffic and other had only



(a) Delay



(b) Jitter

Fig. 2. PIMD vs. FHCF for VBR in Scenario 2. (a) Delay, (b) Jitter.

CBR traffic on them. At a given time, the number of all three types of flows is same. The load is increased by adding one flow of each type. The performance results of the all types of traffic have been plotted.

The simulation results of delay for audio and CBR flows show similar performs of both schedulers (Table II). There difference for a given load is insignificant. The simulation results for jitter were identical and within bounded range (13-17 ms), and hence the data have not been tabulated. In case of the VBR traffic, the delay and jitter using FHCF is lower than PIMD when the load is light. However, the delay and jitter in PIMD remains stable as the load increases, where as FHCF suffers from exceptionally long delay and jitter which makes this algorithm inoperable for real-time services. Hence, the performance of PIMD is much superior to FHCF under heavy load (Fig. 2), and, it can be effectively utilized for mixed traffic of constant and variable bit rate traffic streams.

Table 2. PIMD vs. FHCF in Scenario 2: Delay

No. of Stations	**Audio (ms)**		**CBR (ms)**	
	FHCF	PIMD	***FHCF***	***PIMD***
2	25.5	24.9	21.8	23.1
3	25.1	25.0	21.7	22.7
4	23.3	24.0	22.3	22.9
5	25.4	25.0	22.2	23.5
6	24.8	24.0	21.9	24.3
7	24.7	23.5	24.9	21.9

5 Conclusion

In this paper, we have proposed a computationally simple wireless scheduler for IEEE 802.11e, namely, PIMD wireless scheduler. The scheduler allocates the extra bandwidth in response to the amount of data left in the mobile stations' buffer at the end of allotted transmission opportunity. It has been demonstrated to be stable under heavy load as compared to the other wireless schedulers and it doesn't lower the performance of CBR traffic while allocating dynamic bandwidth for the VBR and bursty traffic. We have proposed an error-free model. The performance of the scheduler under error-prone environment and along with active EDCA has to be analyzed. Work is in progress in the direction of the adding lead-lag model in the scheduler and efficient CAC.

Acknowledgment. This research has partly been sponsored by the Pakistan Telecommunication Cooperation R&D fund under grant R&DF/A-23/2004-05. The first author wants to thank the efforts of his graduate student Mr. Najeeb ul Hassan for his help in editing this paper.

References

1. Wroclawski, J.: The use of RSVP with IETF integrated services. RFC2210 (September 1997)
2. Blake, S., et al.: An architecture for differentiated services. RFC2475 (August 1998)

3. Cao, Y., Li, V.O.K.: Scheduling Algorithms in Broad-Band Wireless Networks. The Proceedings of the IEEE 89(1) (January 2001)
4. Fattah, H., Leung, C.: An Overview of Scheduling Algorithms in Wireless Multimedia Networks. IEEE Wireless Communications (October 2002)
5. IEEE Std 802.11e, Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Amendment 8: Medium Access Control (MAC) Enhancements for Quality of Service (QoS) (November 2005)
6. Nandagopal, T., Lu, S., Bharghavan, V.: A Unified Architecture for the Design and Evaluation of Wireless Fair Queuing Algorithms. In: ACM MOBICOM, pp. 132–142 (August 1999)
7. Grilo, A., Macedo, M., Nunes, M.: A Scheduling Algorithm for QoS Support in IEEE 802.11E Networks. IEEE Wireless Communications, 36–43 (June 2003)
8. Ansel, P., Ni, Q., Turletti, T.: An Efficient Scheduling Scheme for IEEE 802.11e. In: Proceedings of WiOpt, Cambridge, UK (March 2004)
9. Garg, P., Doshi, R., Malek, M., Greene, R., Cheng, M.: Achieving Higher Throughput and QoS in 802.11 Wireless LANs. In: IEEE International Performance Computing and Communications Conference (IPCCC 2003), Phoenix, Arizona (April 2003), `http://nondot.org/~radoshi/cs444n/`
10. Garg, P., Doshi, R., Greene, R., Baker, M., Malek, M., Chang, M.: Using IEEE 802.11e MAC for QoS over Wireless. In: IEEE International Performance Computing and Communications Conference (IPCCC 2003)
11. Dhanakoti, N., Gopalan, S., Vidya, S.: Perfectly Periodic Scheduling for Fault Avoidance in IEEE 802.11e in the Context of Home Networks. ISSRE (2003)
12. Annese, A., Boggia, G., Camarda, P., Grieco, L.A., Mascolo, S.: Providing Delay Guarantees in IEEE 802.11e Networks. In: the 59th IEEE Semi-annual Vehicular Technology Conference, VTC (Spring 2004)
13. Network Simulator Document, `http://www.isi.edu/nsnam/ns/`
14. Ansel, P., Ni, Q., Turletti, T.: FHCF: A Fair Scheduling Scheme for 802.11e WLAN, INRIA Research Report No 4883 (July 2003), `http://www-sop.inria.fr/planete/qni/fhcf`